DEPARTMENT OF COMPUTER SCIENCE
CARLOS III UNIVERSITY OF MADRID

Computer Science

Language Processors

Rules

- The duration of the test is **60 minutes**
- Questions will not be answered during the test
- One cannot re-enter the classroom after leaving it
- The answers must be written using a pen (not a pencil)

1.- Given the grammar:

```
Z ::= bB | cN | j
B ::= NN' | Aj
N' ::= AA' | c
A' ::= c | l
N ::= x
A ::= h | z
```

a) Calculate the FIRST and FOLLOW sets.

FIRST(Z) = {b, c, j}
FIRST(B) = {x, h, z}
FIRST(N') = {c, h, z}
FIRST(A') = {c, λ}
FIRST(N) = {x}
FIRST(A) = {h, z}

FOLLOW(Z) = {\$}
FOLLOW(B) = {\$}
FOLLOW (N') = {\$}
FOLLOW (A') = {\$}
FOLLOW (N) = {h, z, c, \$}
FOLLOW (A) = {c, j, \$}

b) Using the algorithm, determine if it is an LL(1) grammar.

For a grammar to be a LL(1) grammar, it must fulfill that there are not two or more productions in any cell of the analysis table. This condition will occur when:

$\forall$production A ::= α_i|…|α_n:

- $\text{FIRST}(\alpha i) \cap \text{FIRST}(\alpha i) = 0 \ \forall i \neq j$
- If $\alpha i ::= \lambda$ then $\text{FIRST}(\alpha i) \cap \text{FOLLOW}(A) = 0 \ \forall i \neq j$

David Griol Barres, Antonio Berlanga de Jesús, Jesús García Herrero, Juan Manuel Alonso Weber

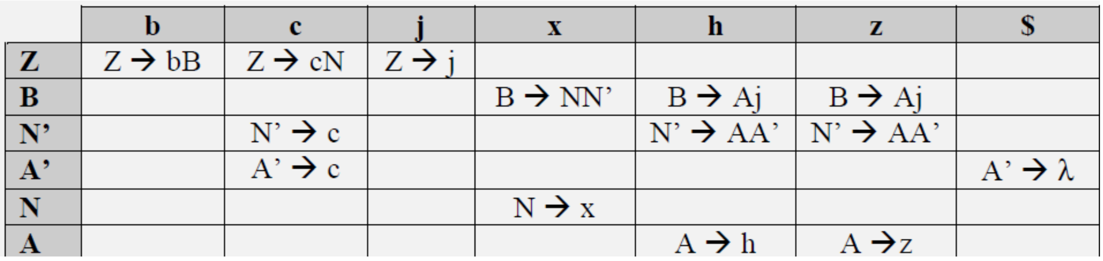
For the given grammar:

FIRST (bB) ∩ FIRST(cN) ∩ FIRST(j) = 0
FIRST (NN’) ∩ FIRST(Aj) = {x} ∩ {h, z} = 0
FIRST (AA’) ∩ FIRST(c) = {h, z} ∩ {c} = 0
FIRST (c) ∩ FIRST(λ) = = 0
FIRST (c) ∩ FOLLOW(A’) {c} ∩ {$} = 0
FIRST (h) ∩ FIRST(z) = = 0

Then, the grammar is an LL(1) grammar.

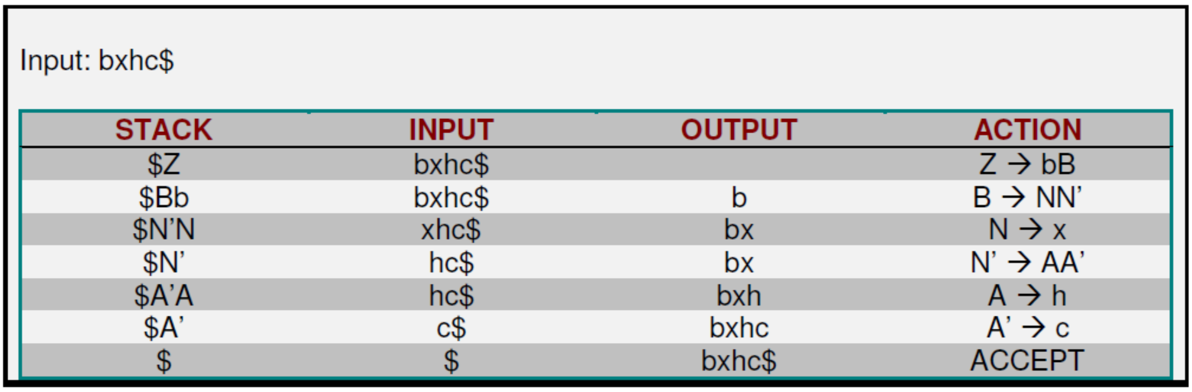
c) Construct the analysis table for the LL(1) table-driven top-down predictive parsing.

The parsing table for the grammar is:



	b	c	j	x	h	z	$
Z	Z → bB	Z → cN	Z → j				
B				B → NN’	B → Aj	B → Aj	
N’		N’ → c			N’ → AA’	N’ → AA’	
A’		A’ → c					A’ → λ
N				N → x			
A					A → h	A →z	

d) Show how the input string bxhc would be analyzed using the parsing table for the grammar.



Input: bxhc$

STACK	INPUT	OUTPUT	ACTION
$Z	bxhc$		Z → bB
$Bb	bxhc$	b	B → NN’
$N’N	xhc$	bx	N → x
$N’	hc$	bx	N’ → AA’
$A’A	hc$	bxh	A → h
$A’	c$	bxhc	A’ → c
$	$	bxhc$	ACCEPT